Pensions Dashboards Programme

Reporting standards

Draft version: November 2022

pensionsdashboardsprogramme.org.uk

Money & Pensions Service

Contents

Introduction

Executive summary

1. The reporting standards ensure that pension providers and schemes (both occupational as well as stakeholder and personal pension schemes connected to the pensions dashboards ecosystem) (pension providers) and qualifying pensions dashboard services (QPDS) understand what information is required to be generated by them and provided to the Pensions Dashboards Programme (PDP) (on behalf of the Money and Pensions Service (MaPS)) for reporting purposes. It also supports monitoring the effectiveness and overall health of the central digital architecture (CDA) and the pensions dashboards ecosystem.
2. The standard ensures there is consistency across both pension providers and schemes and QPDS. When we refer to pension providers and schemes in this standard this includes any contracted third parties.

Background

3. Pensions dashboards are apps, websites or other tools which will help individuals view their pensions information online. They will bring together all an individual's pensions they haven't yet taken, including their State Pension as well as any occupational and personal pensions (including those with an insurer), to support better planning for retirement and growing financial wellbeing.
4. This standard is issued by the Money and Pensions Service (MaPS). MaPS set up the Pensions Dashboards Programme (PDP) in 2019 to design and create the pensions dashboards ecosystem and the supporting governance framework.
5. The pensions dashboards ecosystem contains the central digital architecture (CDA) that will make pensions dashboards work. It will connect millions of individuals to their information across thousands of pensions, via multiple pensions dashboards. For more information about the pensions dashboards ecosystem and its components, see https://www.pensionsdashboardsprogramme.org.uk/ecosystem/.
6. MaPS is also responsible for operating its' own dashboard.
7. Standards are separate from, but designed to complement, the Financial Conduct Authority's (FCA) regulatory framework for pensions dashboard services. Firms which operate a qualifying pensions dashboard service (QPDS) will need to be (or become) FCA authorised, get the regulatory permission to undertake this new regulated activity, and meet any Handbook rules and guidance that the FCA may introduce for firms undertaking this activity.

Purpose

8. This standard covers the requirements on pension dashboards and on pension providers and schemes for generating, recording, and reporting data on a regular basis that will be used alongside data generated within the ecosystem, in order to provide the following:
   - **audit information**
     - **business audit -** information required for non-repudiation purposes, and / or to help us understand what has happened when something goes wrong at the individual user level
   - **ecosystem monitoring information**
     - **protective monitoring -** information including transaction monitoring, (sometimes referred to as 'cyber' monitoring) for security protection and detection
     - **operational monitoring -** information for the purposes of the operational management centre to operate the system and measure its performance in technical terms
   - **ecosystem insight**
     - **management information -** information that allows us to assess how QPDS are being used and whether they are delivering the required service goals, including operational and performance data defined by internal and external stakeholders
     - **oversight reporting -** information that will allow regulators, PDP and other oversight bodies to determine whether PDP, individual pension providers and schemes (pension schemes plus any administrators or integrated service providers carrying out their dashboard functions), and/or individual QPDS's are meeting their obligations
       - this will allow regulators, for instance, to take action against pension schemes and providers that are failing to meet their obligations
9. These types of logging may overlap in terms of data items involved and potentially on purpose across the categories, but it is hoped that the classification is helpful background.

Audience

10. This standard applies to the trustees or managers of occupational pension schemes, the managers of stakeholder and personal pension schemes and insurers (pension providers and schemes) and QPDS's connected to, or are required to connect to, our pensions dashboards ecosystem. However:
    a. due to the connection staging profile the dashboard duties will apply at different dates to different categories of pension provider or scheme
    b. schemes with less than 100 members are exempt unless they voluntarily connect
11. The term dashboard is used interchangeably with QPDS as this may also include the MaPS dashboard even though this standard does not apply to the MaPS dashboard. The MaPS will be adopting it to ensure technical interoperability with the pensions dashboards ecosystem. Where the standard applies only to either pension providers and schemes, or to QPDS, this is highlighted.

12. Third parties (such as administrators or software providers) may apply our standards and guidance on behalf of their pension provider, scheme or QPDS clients. As the standards apply to the pension provider, scheme, or the QPDS, they remain responsible for compliance with them, even if implementation is delegated to a contracted third party.

Jurisdiction

13. This standard applies to all United Kingdom pension providers and schemes subject to the dashboard duties in the Pensions Dashboards Regulations 2022 (the regulations) and FCA regulatory framework.

Other guidance

14. This standard should be read in conjunction with our technical standards, data standards and the code of connection.

Use/evidence

15. Standards are mandatory requirements and, therefore, compliance by pension providers and schemes and QPDS is compulsory.
16. Standards and guidance may be admitted in any proceedings relevant to pension providers and schemes' and QPDS' compliance with their dashboard duties – this also applies to the obligations owed by any other party (for example, a sponsor employer or administrator). It will be the decision of the body hearing the proceedings (including any regulatory proceedings conducted by the FCA or the Pensions Regulator (TPR)) to assess the evidential weight to be attached to any standard or guidance admitted.

Version

17. This is the November 2022 version of the reporting standards. Please refer to the changelog for updates since the last publication.

Summary

18. The following table provides a summarised list of the data feeds that will be required from pension providers and schemes and QPDSs:

Source	API	Satisfies	Event ID	Frequency
pension provider or scheme	business audit view API	business audit	DAPA001	high – each time a dashboard executes a view at the pension provider or scheme
QPDS	business audit dashboard redirect API	business audit protective monitoring	DASA001	high – each time a user re-directs to C&A
pension provider or scheme	data provider health check API	operational monitoring	DAPO001	high - every 60s per pension provider or scheme
QPDS	dashboard access API	protective monitoring	DASP001	near real time
QPDS	view requests API	MI reporting	DASM001	low – daily
QPDS	dashboard access API	MI reporting	DASM002	low – daily
pension provider or scheme	data provider availability API	MI reporting	DAPM001	low – daily
QPDS	dashboard provider availability API	MI reporting	DASM003	low – daily
pension provider or scheme	data provider value data unavailable API	MI reporting	DAPM002	low – daily

Record keeping requirement

Audit

19. Both QPDS and pension providers and schemes shall keep audit logs including transaction identifiers for all ecosystem interactions for 6 years. They should be ready to provide mediated access to their internal audit records. In the event of investigation of a customer complaint or security incident, or any activity requiring regulatory (or other) investigation. These internal audit records should also contain any unique transaction identifiers supplied to, or received from, the

CDA, or QPDS/pension provider or scheme to ensure no individual user's details are disclosed (the standard for these transaction identifiers can be found in the technical standards).

Complaints

20. Both QPDS and pension providers and schemes should keep records of complaints information related to pensions dashboards for 6 years. This should include volume, nature and outcome information and the records should be accessible to PDP, upon request.

Pension owner satisfaction

21. QPDS shall make available to PDP survey data from any surveys they undertake to support overall policy evaluation.

Pension owner behaviour

22. QPDS shall make available to PDP data they collect on pension owner behaviour (ie drop off rates, after finding a pension do pensions owners go onto view) from their own analytics regarding the pensions dashboard service.

Generic API information

23. This section details information which applies across all of the different 'reporting' domains.

Response codes

24. The following HTTP response codes will be used:

- 202 - the request has been accepted for processing, but the processing has not been completed
- 400 - the server can't process the request due to something that is perceived to be a client error, eg incorrect message format or missing certificate (for example, malformed request syntax, invalid request message framing, or deceptive request routing)
- 401 - the client request has not been completed because it lacks valid authentication credentials for the requested resource eg missing certificate
- 429 - too many requests indicates the user has sent too many requests in a given amount of time (rate limiting)

- 5xx - these errors are caused by the server being unable to fulfill an apparently valid request, an indication about the nature of the error should be included

Retry processing

25. In the event of a HTTP response returning anything other than an HTTP 202, it is the responsibility of the external system to implement a retry mechanism for any messages which failed to deliver successfully. It is recommended that a circuit breaker pattern or other back off strategies are implemented in this scenario.

Paths

26. The following paths will be used for the APIs as appropriate:

- /events/audit
- /events/operational
- /events/protective
- /events/MI

27. The individual reporting sections will indicate the specific path to be used.

HTTP requests

28. For all 'reporting' APIs – HTTP POST will be used.
29. The header will also contain the X-Request-ID a universally unique identifier (uuid) which will be created by the pension provider or scheme and which will be used to uniquely identify the audit event in question.

Hosting

30. All APIs will be hosted on the authorisation server, which is part of the consent and authorisation (C&A) service.

Authentication

31. The client(s) (dashboards and pension providers and schemes) will authenticate themselves with the authorisation server at run time as defined in rfc8705 section 2. In addition to this, all endpoint connections are secured using mutual TLS.

Business audit

Overview

32. A business audit event is the recording of an event of significance. Its primary purpose is to ensure the creation of records which assist in the non-repudiation of those events. This enables proof of users' actions and the causal relationships between users when events take place.

33. Business audit events are those which:
    - release data from one domain to another and must be recorded
    - lead to the persistence of data, or its modification
    - enable control of, or dependency, between events across domains

34. External users, the human users are *pension owners and their delegates*, the system users are the CDA, dashboards, pension providers and schemes, and identity services.

35. For pension providers and schemes and QPDS the events that they must provide for business audit are detailed in the sections below.

Business audit view API

36. The business audit view API is to provide an audit trail for view requests and is initiated by the pension provider(s) or scheme(s).

Frequency

37. Must be created immediately after pension details are returned to the requesting dashboard provider as a result of a view API request.

Data definitions

Data item	Description	Type	Format	Conditionality
event_type	**alphanumeric** code used to identify the event DAPA001 where 'DAP' – data provider (pension provider or scheme), 'A' – audit, '001' – numerical identifier for this event type	string	source + reporting type + event type	mandatory

Data item	Description	Type	Format	Conditionality
view_request_id	the view_request_id in the body of the message is a uuid which is created by the QPDS and which will have been sent to the pension provider or scheme as part of the originating View request API message standards for the Request_id generation are detailed in the technical standards	string	UUID	mandatory
holder_name	holder name of the scheme or part scheme the view was initiated against	string	UUID	mandatory
date_ts	date + time of the event	string	date-time ISO 8601	mandatory

Example

38. POST /events/audit HTTP/1.1

```
Content-Type: application/json;charset=UTF-8

X-Request-ID: fbb0ea43-c231-4449-bac1-c3f1b751547f

  {
   "event_type": "DAPA001",
   "view_request_id": "ce08b849-c8aa-49ec-9f94-3d0e6eb7ef9c",
   "holder_name": "c43ab8b9-8a82-4687-ab93-877b85012cd1",
   "date_ts": "2022-11-22T01:23:45+05:00"
  }
```

Business audit dashboard redirect API

39. The business audit dashboard redirect API is initiated by the QPDS(s) and will be created as a result of redirect actions from the dashboard to the CDA.

Frequency

40. Must be created immediately after a pension owner is redirected to the CDA.

Data definitions

Data item	Description	Type	Format	Conditionality
event_type	**alphanumeric** code used to identify the event. DASA001 where 'DAS' – dashboard provider, 'A' – audit, '001' – numerical identifier for this event type	string	source + reporting type + event type	mandatory
reason_code	reason for the redirect values: **find -** when redirected to find a pension **consent** – when redirected to manage consent **chatbot** – when redirected to chatbot **complain** – when redirected to complain	string		mandatory
date_ts	date + time of the event	string	date-time ISO 8601	mandatory

Example

41. POST /events/audit HTTP/1.1

```
Content-Type: application/json;charset=UTF-8

X-Request-ID: fbb0ea43-c231-4449-bac1-c3f1b751547f

{
    "event_type": "DASA001",
    "reason_code": "find",
    "date_ts": "2022-11-22T01:23:45+05:00"
```

}

Operational monitoring

Overview

42. Operational monitoring provides metrics and alerts, which enable the reliable operation and management of the service by the CDA's operational management centre.

Data provider health check API

43. The data provider health check API is initiated by the pension provider(s) and scheme(s) and will send regular health checks to the C&A service related to the health of the find and view endpoints.

Frequency

44. Every 60 seconds

Data definitions

Data item	Description	Type	Format	Conditionality
event_type	**alphanumeric** code used to identify the event DAPO001 where 'DAP' – data provider (pension provider or scheme), 'O' – operational monitoring, '001' – numerical identifier for this event type	string	source + reporting type + event type	mandatory
endpoint_type	purpose of endpoint values: **find** – find endpoint **view** – view endpoint	string		mandatory
endpoint_url	URL or address of the endpoint	string	URL	mandatory

Data item	Description	Type	Format	Conditionality
Not_ok	holder name of the scheme or part scheme that is not available an empty “Not_ok” array indicates that all holder_name(s) within an endpoint are operating as expected	string array	UUID	conditional
date_ts	date + time of the event	string	date-time ISO 8601	mandatory

Example

45. POST /events/operational HTTP/1.1

```
        Content-Type: application/json;charset=UTF-8

        X-Request-ID: fbb0ea43-c231-4449-bac1-c3f1b751547f

         {

            "event_type": "DAPO001",

            "endpoint_type": "find",

            "endpoint_url": "https://testep.com",

            "Not_ok": [

              "a911bbd8-459f-4a14-ba10-2a31699294a5",

              "cb4ec6a1-d45a-4be1-829a-43d97ec23c17"

            ],
         "date_ts": "2022-11-22T01:23:45+05:00"   }
```

Protective monitoring

Overview

46. Protective monitoring, transaction monitoring, 'cyber' monitoring, enable prevention or detection of incidents which relate to the secure operation of the service.

47. This helps the CDA to identify attack indicators, and looks for indicators of compromise, detecting events and raising incidents to be handled by the CDA's security operations centre (SOC).

Dashboard access API

48. The dashboard access API is initiated by the QPDS(s) at the point that the dashboard itself is accessed by a customer. The purpose of capturing this protective monitoring data from the dashboard is to enable downstream security checks to be performed within CDA.

Frequency

49. Must be created immediately after a pension owner accesses the dashboard

Data definitions

Data item	Description	Type	Format	Conditionality
event_type	**alphanumeric** code used to identify the event DASP001 where 'DAS' – dashboard provider, 'P' – protective monitoring, '001' – numerical identifier for this event type	string	source + reporting type + event type	mandatory
ip	IP address	string	IPV4	mandatory
browser_fp	details of the browser used	object		mandatory

Data item	Description	Type	Format	Conditionality
browser_hash	the bowser hash schema is used in fraud detection and cybersecurity, a browser hash helps to understand who you are dealing with based on the technology they use	string		mandatory
screen_width	width of screen used to access the dashboard	string		mandatory
screen_height	height of screen used to access the dashboard	string		mandatory
screen_depth	depth of screen used to access the dashboard	string		mandatory
platform	platform used i.e. widows 32, OS etc	string		mandatory
hware_concurrency	number of concurrent threads possible	string		mandatory
Device_mem_gb	device memory in gigabytes	string		mandatory
date_ts	date + time of the event	string	date-time ISO 8601	mandatory

Example

50. POST /events/protective HTTP/1.1

```
Content-Type: application/json;charset=UTF-8

X-Request-ID: fbb0ea43-c231-4449-bac1-c3f1b751547f

   {
    "event_type": "DASP001",
    "ip": "192.168.0.10",
    "browser_fp": {
      "browser_hash": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/103.0.5060.114 Safari/537.36 Edg/103.0.1264.49",

      "screen_width": "1536",
      "screen_height": "864",
      "screen_depth": "24",
      "platform": "Win32",
      "hware_concurrency": "8",
      "Device_mem_gb": "8"
    },
    "date_ts": "2022-06-29T01:23:45+00:00"
  }
```

Ecosystem oversight and insight (MI)

Overview

51. Oversight information (including operational, financial, and performance data) to enable an oversight body (FCA, TPR, DWP) to manage compliance and non-compliance of regulated entities and PDP to have oversight of participants behavior. Additionally, this allows PDP to assess how the pensions dashboards service is being used and whether it is delivering the required service.

View requests API

52. The view request API is initiated by the QPDS(s). The purpose is to capture ***information*** related to view requests made to the pension provider(s) or scheme(s).

Frequency

53. This reporting event will be sent between 00:00 and 06:00 on the day following when the events themselves occurred.

54. Stats will be provided on a daily summarised basis, as opposed to sending underlying individual events. The start_date and end_date will be the same date to signify a daily reporting period.

.

Data definitions

Data item	Description	Type	Format	Conditionality
event_type	**alphanumeric** code used to identify the event DASM001 where 'DAS' – dashboard provider, 'M' – MI, '001' – numerical identifier for this event type	string	source + reporting type + event type	mandatory
view_stats	statistics for view	string array		mandatory
holder_name	holder name of the scheme or part scheme that the view stats relate to	string	UUID	mandatory
send_count	number of view requests for the holder name above	integer		mandatory
ave_resp_time	'ave_response_time' is the average time in milliseconds between the original request being sent to the provider and the pension details being returned to the dashboard	integer		mandatory
non_ack_count	number of times a nack was sent from the view endpoint from the holder name above	integer		mandatory

Data item	Description	Type	Format	Conditionality
sla_fail_count	number of times the view response was outside SLA from the holder name above	integer		mandatory
http_resp_stats	IP address	string array		mandatory
http_resp	HTTP response type i.e. 200 for a successful response 400 for an unsuccessful response and holder name	string		mandatory
count	number of HTTP response types per type and holder name	integer		mandatory
start_date	date	string	date	mandatory
end_date	date	string	date	mandatory
date_ts	date + time of the event	string	date-time ISO 8601	mandatory

Example

55. POST /events/MI HTTP/1.1

```
Content-Type: application/json;charset=UTF-8

X-Request-ID: fbb0ea43-c231-4449-bac1-c3f1b751547f

     {

   "event_type": "DASM001",

   "view_stats": [
        {
          "holder_name": "e0d1d7ce-4fbb-4303-a0cc-1c8d82e23682",
          "send_count": 1200,
          "ave_resp_time": 1672,
          "non-ack-count": 50,
          "sla_fail_count": 30,
          "http_resp_stats": [
            {
              "http_resp": "200",
              "count": 1000
            }
          ]
        }
      ],
      "start_date": "2022-06-29",
      "end_date": "2022-06-29",
      "date_ts": "2022-11-22T01:23:45+05:00"
    }
  ]
}
```

Dashboard access API

56. The dashboard access API is initiated by the QPDS(s). The purpose is to capture the MI related to user access at a dashboard level.

Frequency

57. This reporting event will be sent between 00:00 and 06:00 on the day following when the events themselves occurred.

58. Stats will be provided on a daily summarised basis, as opposed to sending underlying individual events. The start_date and end_date will be the same date to signify a daily reporting period.

Data definitions

Data item	Description	Type	Format	Conditionality
event_type	**alphanumeric** code used to identify the event DASM002 where 'DAS' – dashboard provider, 'M' – MI, '002' – numerical identifier for this event type	string	source + reporting type + event type	mandatory
access_stats	Statistics for dashboard provider	string array		mandatory
login_count	number of pension owners accessing pensions dashboards via a QPDS account	string		mandatory
guest_count	number of pension owners accessing pensions dashboards via a guest access	string		mandatory
start_date	date of the event	string	date	mandatory
end_date	date of the event	string	date	mandatory
date_ts	date + time of the event	string	date-time ISO 8601	mandatory

Example

59. POST /events/MI HTTP/1.1

```
Content-Type: application/json;charset=UTF-8

X-Request-ID: fbb0ea43-c231-4449-bac1-c3f1b751547f

    {
```

```
  "event_type": "DASM002",
  "access_stats": {

    "login_count": 1200,

    "guest_count": 0

  },

  "start_date": "2022-06-30",

  "end_date": "2022-06-30",

  "date_ts": "2022-06-29T01:23:45+00:00"

}
```

Data provider availability API

60. The purpose of this API is to capture statistics in regard to the availability of a pension provider or scheme.

Frequency

61. This reporting event will be sent between 00:00 and 06:00 on the day following when the events themselves occurred.

62. Stats will be provided on a daily summarised basis, as opposed to sending underlying individual events. The start_date and end_date will be the same date to signify a daily reporting period.

.

Data definitions

Data item	Description	Type	Format	Conditionality
event_type	**alphanumeric** code used to identify the event DAPM001 where 'DAP' – data provider (pension provider or scheme), 'M' – MI, '001' – numerical identifier for this event type	string	source + reporting type + event type	mandatory
avail_stats	statistics for pension provider or scheme	string array		mandatory
holder_name	holder name of the scheme or part scheme that the availability stats relate to	string	UUID	mandatory
find_ep_avail	find availability per scheme or part scheme availability will be reported in seconds with the 100% availability being reported as 86400 (1 day)	string	seconds	mandatory
view_ep_avail	view availability per scheme or part scheme availability will be reported in seconds with the 100% availability being reported as 86400 (1 day)	string	seconds	mandatory
start_date	date of the event	string	date	mandatory
end_date	date of the event	string	date	mandatory

Data item	Description	Type	Format	Conditionality
date_ts	date + time of the event	string	date-time ISO 8601	mandatory

Example

63. POST /events/MI HTTP/1.1

```
Content-Type: application/json;charset=UTF-8

X-Request-ID: fbb0ea43-c231-4449-bac1-c3f1b751547f

  {

    "event_type": "DAPM001",

    "avail_stats": [

      {

        "holder_name": "6aa593b9-831a-4dd1-bbdc-cb6150fb232e",

        "find_ep_avail": "86400",

        "view_ep_avail": "86400"

      },

      {

        "holder_name": "126d1135-33e4-4e95-baee-da24d65027f5",

        "find_ep_avail": "86400",

        "view_ep_avail": "23400"

      }
    ],

    "start_date": "2022-07-01",

    "end_date": "2022-07-01",

    "date_ts": "2022-07-01T01:23:45+00:00"

  }
```

Dashboard provider availability API

64. The purpose of this API is to capture statistics in regard to the availability of a ***dashboard*** provider.

Frequency

65. This reporting event will be sent between 00:00 and 06:00 on the day following when the events themselves occurred.
66. Stats will be provided on a daily summarised basis, as opposed to sending underlying individual events. The start_date and end_date will be the same date to signify a daily reporting period.

.

Data definitions

Data item	Description	Type	Format	Conditionality
event_type	**alphanumeric** code used to identify the event DASM003 where 'DAS' – dashboard provider, 'M' – MI, '003' – numerical identifier for this event type	string	source + reporting type + event type	mandatory
dashboard_avail	dashboard availability availability will be reported in seconds with the 100% availability being reported as 86400 (1 day)	string	seconds	mandatory

Data item	Description	Type	Format	Conditionality
start_date	date of the event	string	date	mandatory
end_date	date of the event	string	date	mandatory
date_ts	date + time of the event	string	date-time ISO 8601	mandatory

Example

67.

```
POST /events/MI HTTP/1.1

      Content-Type: application/json;charset=UTF-8

      X-Request-ID: fbb0ea43-c231-4449-bac1-c3f1b751547f

        {

          "event_type": "DASM003",

          "dashboard_avail": "86400",

          "start_date": "2022-07-01",

          "end_date": "2022-07-01",

          "date_ts": "2022-07-01T01:23:45+00:00"

        }
```

Data provider value data unavailable API

68. The purpose of this API is to capture statistics in regard to the availability of value data.

Frequency

69. This reporting event will be sent between 00:00 and 06:00 on the day following when the events themselves occurred.

70. Stats will be provided on a daily summarised basis, as opposed to sending underlying individual events. The start_date and end_date will be the same date to signify a daily reporting period.

Data definitions

Data item	Description	Type	Format	Conditionality
event_type	**alphanumeric** code used to identify the event DAPM002 where 'DAP' – data provider (pension provider or scheme), 'M' – MI, '002' – numerical identifier for this event type	string	source + reporting type + event type	mandatory
eri_unavail	where estimated retirement income (eri) values are unavailable on a view request	String array		mandatory
eri_unavail_reason	reason code as defined in the data standard for unavailable eri	String		mandatory
eri_unavail_stats	on a view request where estimated retirement income (eri) values are unavailable	String		mandatory
eri_exceed_stats	for each calculation, a single report where the pension provider or scheme is not ready to provide the value information within the deadline set out in the regulations	String array		conditional
acc_unavail	where accrued pension (acc) values are unavailable	String array		mandatory
acc_unavail_reason	reason code as defined in the data standard for unavailable acc	String		mandatory

Data item	Description	Type	Format	Conditionality
acc_unavail_stats	on a view request where the accrued pension (acc) values are unavailable	String		mandatory
acc_exceed_stats	for each calculation, a single report where the pension provider or scheme is not ready to provide the value information within the deadline set out in the regulations	String		mandatory
start_date	date of the event	string	date	mandatory
end_date	date of the event	string	date	mandatory
date_ts	date + time of the event	string	date-time ISO 8601	mandatory

Example

```
71.  POST /events/MI HTTP/1.1

           Content-Type: application/json;charset=UTF-8

           X-Request-ID: fbb0ea43-c231-4449-bac1-c3f1b751547f

              {

                "event_type": "DAPM002",

                ”eri_unavail”:[

                {

                     ”eri_unavail_reason”:”DBC”,

                      ”eri_unavail_stats”:”1”,
                      ”eri_exceed_stats”:”2”
              }
             ]
               ”acc_unavail”:[
                {
                      ”acc_unavail_reason ”:”NEW”,
                      ”acc_unavail_stats”:”10”,
                      ”acc_exceed_stats”0”
               }
               ]
                "start_date": "2022-07-01",
                "end_date": "2022-07-01",
                "date_ts": "2022-07-01T01:23:45+00:00"
              }
```